Diagnosing Diabetes Type II Using a Soft Intelligent Binary Classification Model

Mehdi Khashei *1, Saeede Eftekhari 2, Jamshid Parvizian 3

Department of Industrial Engineering, Isfahan University of Technology (IUT)

Isfahan University of Technology (IUT), Isfahan 84156-83111, Iran

*1Khashei@in.iut.ac.ir; 2s.eftekhari@in.iut.ac.ir; 3japa@cc.iut.ac

Abstract

Diabetes, named also silent killer, is a metabolic disease characterized by high blood glucose levels, which result from body does not produce enough insulin or the body is resistant to the effects of insulin. Classification models are one of the most widely used groups of data mining tools that greatly help physicians to improve their prognosis, diagnosis or treatment planning procedures. Classification accuracy is one of the most important features in order to choose the appropriate classification model; hence, the researches directed at improving upon the effectiveness of these models have never stopped. Nowadays, despite the numerous classification models proposed in several past decades, it is widely recognized that diabetes are extremely difficult to classify. In this paper, a hybrid binary classification model is proposed for diabetes type II classification, based on the basic concepts of soft computing and artificial intelligence techniques. Empirical results of Pima Indian diabetes data classification indicate that hybrid model is generally better than other linear/nonlinear, soft/hard, and classic/intelligent classification models presented for diabetes classification. Therefore, our proposed model may be a suitable alternative model for medical classification to achieve greater accuracy, and to improve medical diagnosis.

Keywords

Artificial Intelligence; Soft Computing; Classification; Medical Diagnosis; Diabetes

Introduction

Diabetes mellitus has become a general chronic disease that affects between 2% and 4% of the global population, and its avoidance and effective treatment are undoubtedly crucial public health and health economics issues in the 21st century. Diabetes is a metabolic diseases characterized by high blood glucose levels, which result from body does not produce enough insulin or the body is resistant to the effects of insulin, named silent killer. The body needs insulin to use sugar, fat and protein from the diet for energy. Diabetes is associated with many complications and it can increase the risk of blindness, blood pressure, heart disease, kidney disease, and nerve damage (Temurtas et al., 2009).

Diabetes disease is generally categorized in two categories, diabetes type I and diabetes type II. The most usual form of diabetes is diabetes type II or diabetes mellitus type. In diabetes type II the body is resistant to the effects of insulin. Millions of people have been diagnosed with diabetes type II and unfortunately, many more unaware that they are at high risk. Despite recent medical progresses, early diagnosis of disease has improved but about half of the patients of diabetes type II are unaware of their disease and may take more than ten years as the delay from disease onset to diagnosis while early diagnosis and treatment of this disease is vital.

Classification systems have been widely utilized in medical domain to explore patient's data and extract a predictive model. They help physicians to improve their prognosis, diagnosis or treatment planning procedures. In recent years, many studies have been performed in the diagnosis of diabetic disease literature. Several different methods, such as logistic regression, Naive Bayes, Semi-Naive Bayes, multi-layer perceptrons (MLPs), radial basis functions (RBFs), general regression neural networks (GRNNs), support vector machines (SVMs), Least square support vector machines (LS-SVMs) have been used in some of these studies (Charya et al., 2010; Kayaer & Yildirim, 2003; Bennett & Blue, 1998; Friedman et al. 1997). Decision tree techniques also have been widely used to build classification models as such models closely resemble human reasoning and are easy to understand.

Ton *et al.* (2006) constructed a classification model for diabetes type II using anthropometrical body surface scanning data. They applied four data mining approaches, including artificial neural network, decision tree, logistic regression, and rough set, to select the relevant features from the data to classify diabetes. They showed that the accuracy of the decision tree and rough set was found to be superior to that of logistic regression and neural network. Joseph *et al.* (2002) used the classification tree for classification and regression trees with a binary target and ten attributes including age, sex, emergency department visits, office visits, comorbidity index, dyslipidemia, hypertension, cardiovascular disease, retinopathy and end-stage renal disease.

Polat *et al.* (2008) proposed a new cascade learning system based on generalized discriminant analysis and least square support vector machine for classification of diabetes disease. They examined the robustness of their proposed system by using classification accuracy, k-fold cross-validation method and confusion matrix. Huang *et al.* (2007) first applied feature selection methods in order to discover key attributes affecting diabetic control, and then used three complementary classification techniques including Naive Bayes, IB 1and C4.5, to classify how well the patients' condition was controlled. Hung *et al.* (2012) proposed system utilized the supervised classifier to screen the import risk factors for different chronic illnesses and then used these significant risk factors to implement the classification and to construct the early-warning criteria.

Calisir and Dogantekin (2011) introduced an automatic diagnosis system integrated linear discriminant analysis (LDA) and Morlet wavelet support vector machine (LDA–MWSVM) for diabetes classification. Patil *et al.* (2010) build a hybrid classification model, which could accurately classify newly diagnosed patients (pregnant women) into either group that is likely to develop diabetes or into a group, which does not develop the diabetes in five years from the time of first diagnosis. Zhao (2007) propose a multi-objective genetic programming approach to developing Pareto optimal decision trees and illustrate its application in the diabetes classification.

Recently, fuzzy approaches have become one of the well-known solutions for improving classification models. Fuzzy theory was originally developed to deal with problems involving linguistic terms (Zadeh, 1975a) and have been successfully applied to the broad range of problems. Fuzzy logic (Zadeh, 1975b) improves classification and decision support systems (DSS) by allowing the use of overlapping class definitions and its powerful capabilities to handle uncertainty and vagueness (Shi et al., 1999). Ganji & Abadeh (2011) proposed an ant colony-based classification system to extract a set of fuzzy rules for diagnosis of diabetes disease, named FCS-ANTMINER. Khashei *et al.* (2012) proposed new hybrid model combining artificial intelligence with fuzzy models in order to benefit from unique advantages of these techniques to construct an efficient and accurate hybrid classifier. Kahramanli and Allahverdi (2008) presented a new method for classification of data of a medical database and developed a hybrid neural network that includes artificial neural networks and fuzzy neural networks (FNNs).

In this paper, a two-stage hybrid classification model of traditional multi-layer perceptrons is proposed in order to yield more accurate results than other those models in diabetes type II classification. In the first stage of proposed model, a multi-layer perceptron is used to pre-process of raw data and provide necessary background in order to apply a fuzzy regression model. In second stage, the obtained parameters of first stage are considered in the form of fuzzy numbers and then the optimum values of proposed model parameters are calculated using the basic concept of fuzzy regression. In order to show the effectiveness and appropriateness of proposed model, its performance is compared with those of some fuzzy and nonfuzzy, linear and nonlinear, and intelligent classification models. Empirical results of Pima Indian diabetes data classification indicate that the proposed model is an effective way in order to improve classification accuracy.

The rest of the paper is organized as follows. In the next Section, the basic concepts and modelling approaches of the traditional multi-layer perceptrons (MLPs) and other used classification models in this paper are briefly reviewed. The formulation of the hybrid proposed model to classification tasks is reviewed in Section 3. In Section 4, the used data set, Pima Indian diabetes data set, is briefly introduced. In Section 5, the proposed model is applied to Pima Indian diabetes data set classification. In Section 6, the performance of the proposed model is compared to some other classification models, presented in the literature for diabetes classification. Finally, the conclusions are discussed.

Classification Approaches

In this section, the basic concepts and modelling approaches of the multi-layer perceptrons (MLPs), support vector machines (SVMs), *K*-nearest neighbour

(KNN), quadratic discriminant analysis (QDA), and linear discriminant analysis (LDA) models for classification are briefly reviewed.

Multi-Layer Perceptrons (MLPs)

Artificial neural networks (ANNs) are computer systems developed to mimic the operations of the human brain by mathematically modelling its neuro-physiological structure. Artificial neural networks have been shown to be effective at approximating complex nonlinear functions (Zhang, 2001). For classification tasks, these functions represent the shape of the partition between classes. In artificial neural networks, computational units called neurons replace the nerve cells and the strengths of the interconnections are represented by weights, in which the learned information is stored. This unique arrangement can acquire some of the neurological processing ability of the biological brain such as learning and drawing conclusions from experience. Artificial neural networks combine the flexibility of the boundary shape found in *K*-nearest neighbour with the efficiency and low storage requirements of discriminant functions. Like the K-nearest neighbour, artificial neural networks are data driven; there are no assumed model characteristics or distributions, as is the case with discriminant analysis (Berardi & Zhang, 1999).

Multi-layer perceptrons (MLPs) are one of the most important and widely used forms of artificial neural networks for modelling, forecasting, and classification (Silva, 2008). These models are characterized by the network of three layers of simple processing units connected by acyclic links (Fig. 1). The relationship between the output (y) and the inputs $(x_1,x_2,...,x_p)$ has the following mathematical representation:

$$y_t = w_0 + \sum_{j=1}^{q} w_j \cdot g(w_{0,j} + \sum_{i=1}^{p} w_{i,j} \cdot x_{t,i}) + \varepsilon_t , \qquad (1)$$

where, $w_{i,j}(i = 0,1,2,...,p,\ \ j = 1,2,...,q)$ and $w_j(j = 0,1,2,...,q)$ are model parameters often called connection weights; g is the hidden transfer function; ε_t is the white noise time t; p is the number of input nodes; and q is the number of hidden nodes. Data enters the network through the input layer, moves through hidden layer, and exits through the output layer. Each hidden layer and output layer node collects data from the nodes above it (either the input layer or hidden layer) and applies an activation function. Activation functions can take several forms. The type of activation function is indicated by the situation of the neuron within the network. In the majority of cases input layer neurons do not have an activation function, as their role is to transfer the inputs to the hidden layer. The logistic and hyperbolic functions are often used as hidden layer and output transfer functions for classification problems that are shown in Eq. 2 and Eq. 3, respectively. Other transfer functions can also be used such as linear and quadratic, each with a variety of modelling applications.

$$Sig(x) = \frac{1}{1 + exp(-x)}. \qquad (2)$$

$$Tanh(x) = \frac{1 - exp(-2x)}{1 + exp(-2x)}. \qquad (3)$$

The simple network given by (1) is surprisingly powerful in that it is able to approximate the arbitrary function as the number of hidden nodes when q is sufficiently large. In practice, simple network structure that has a small number of hidden nodes often works well in out-of-sample forecasting. This may be due to the over-fitting effect typically found in the neural network modelling process. An over-fitted model has a good fit to the sample used for model building but has poor generalizability to data out of the sample.

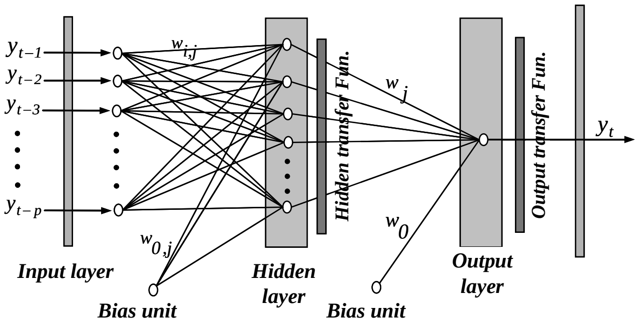

FIG. 1 MULTI-LAYER PERCEPTRON STRUCTURE $(N^{(p-q-1)})$

There exist many different approaches such as the pruning algorithm, the polynomial time algorithm, the canonical decomposition technique, and the network information criterion for finding the optimal architecture of an artificial neural network. These approaches can be generally categorized as follows (Khashei & Bijari, 2010): (i) Empirical or statistical methods that are used to study the effect of internal parameters and choose appropriate values for them based on the performance of model. The most systematic and general of these methods utilizes the principles from Taguchi's design of experiments. (ii) Hybrid methods such as fuzzy inference where the artificial neural network can be interpreted as an adaptive fuzzy system or it can operate on fuzzy instead of real numbers. (iii) Constructive and/or pruning algorithms that, respectively, add and/or remove neurons from an initial architecture using a previously specified criterion to indicate how artificial neural network performance is affected by the changes. The basic rules are that neurons are added when training is slow or when the mean squared error is larger than a specified value. In opposite, neurons are removed when a change in a neuron's value does not correspond to a change in the network's response or when the weight values that are associated with this neuron remain constant for a large number of training epochs. (iv). Evolutionary strategies that search over topology space by varying the number of hidden layers and hidden neurons through application of genetic operators and evaluation of the different architectures according to an objective function (Benardos et al. 2007).

Although many different approaches exist in order to find the optimal architecture of an artificial neural network, these methods are usually quite complex in nature and are difficult to implement (Zhang & Patuwo, 1998). Furthermore, none of these methods can guarantee the optimal solution for all real forecasting problems. To date, there is no simple clear-cut method for determination of these parameters and the usual procedure is to test numerous networks with varying numbers of hidden units, estimate generalization error for each and select the network with the lowest generalization error (Hosseini et al. 2006). Once a network structure is specified, the network is ready for training a process of parameter estimation. The parameters are estimated such that the cost function of neural network is minimized. Cost function is an overall accuracy criterion such as the following mean squared error:

$$E = \frac{1}{N}\sum_{n=1}^{N}(e_i)^2 = \\ = \frac{1}{N}\sum_{n=1}^{N}\left(y_t - \left(w_0 + \sum_{j=1}^{q} w_j \, g(w_{0j} + \sum_{i=1}^{p} w_{i,j} \, y_{t-i}) \right) \right)^2 , \qquad (4)$$

where, N is the number of error terms. This minimization is done with some efficient nonlinear optimization algorithms other than the basic back propagation training algorithm (Rumelhart & McClelland, 1986), in which the parameters of the neural network, $w_{i,j}$, are changed by an amount $\Delta w_{i,j}$, according to the following formula:

$$\Delta w_{i,j} = -\eta \frac{\partial E}{\partial w_{i,j}}, \qquad (5)$$

where, the parameter η is the learning rate and $\partial E / \partial w_{i,j}$ is the partial derivative of the function E with respect to the weight $w_{i,j}$. This derivative is commonly computed in two passes. In the forward pass, an input vector from the training set is applied to the input units of the network and is propagated through the network, layer by layer, producing the final output. During the backward pass, the output of the network is compared with the desired output and the resulting error is then propagated backward through the network, adjusting the weights accordingly. To speed up the learning process, while avoiding the instability of the algorithm, Rumelhart and McClelland (1986) introduced a momentum term δ in Eq. (5), thus obtaining the following learning rule:

$$\Delta w_{i,j}(t+1) = -\eta \frac{\partial E}{\partial w_{i,j}} + \delta \, \Delta w_{i,j}(t), \qquad (6)$$

The momentum term may also be helpful to prevent the learning process from being trapped into poor local minima, and it is usually chosen in the interval [0; 1]. Finally, the estimated model is evaluated using a separate hold-out sample that is not exposed to the training process.

Linear Discriminant Analysis (LDA)

Linear discriminant analysis (LDA) is a very simple and effective supervised classification method with wide applications. The basic theory of linear discriminant analysis is to classify compounds by dividing an n-dimensional descriptor space into two regions that are separated by a hyper-plane that is defined by a linear discriminant function. Discriminant analysis generally transforms classification problems into functions that partition data into classes, thus reducing the problem to the identification of a function. The focus of discriminant analysis is on determining this functional form and estimating its coefficients. In the linear discriminant analysis, this function is assumed to be linear. Ronald Aylmer Fisher (1936) first introduced the linear discriminant function. Fisher's linear discriminant function works by finding the mean of the set of attributes for each class and using the mean of these means as the boundary. The function achieves this by projecting the attribute points onto the vector that maximally separates their class means and minimizes their within-class variance. The Fisher's linear discriminant function can be written as follows:

$$X'S^{-1}\left(\bar{X}_2-\bar{X}_1\right)-{}^{1}\!/_{2}\left(\bar{X}_2+\bar{X}_1\right)'S^{-1}\left(\bar{X}_2-\bar{X}_1\right)>c \qquad (7)$$

where X is the vector of the observed values, $\bar{X}_i\,(i=1,2)$, is the mean of values for each group, S is the sample covariance matrix of all variables, and c is the cost function. If the misclassification cost of each group is considered equal, c is set to zero. A member is classified into one group if the result of the equation is greater than c (or zero) and into the other if less than c (or zero). A result equal to c indicates that a sample cannot be classified into either class based on the features used in the analysis.

The linear discriminant function distinguishes between two classes. If a data set has more than two classes, the process must be broken down into multiple two-class problems. The linear discriminant function was found for each class versus all samples that were not of that class (one-versus-all). Final class membership for each sample was determined by the linear discriminant function that produced the highest value. Linear discriminant analysis is optimal when the variables are normally distributed with equal covariance matrices. In this case, the linear discriminant function is in the same direction as the Bayes optimal classifier (Billings & Lee, 2002). The linear discriminant is known to perform well on moderate sample sizes when compared to more complex methods (Ghiassi & Burnley, 2010). As a straightforward mathematical function, requiring nothing more complicated than matrix arithmetic, the linear discriminant is relatively simple to perform. The assumption of linearity in the class boundary, however, limits the scope of application for linear discriminant analysis. Real-world data frequently cannot be separated by a linear boundary. When boundaries are nonlinear, the performance of the linear discriminant may be inferior to other classification methods.

Quadratic Discriminant Analysis (QDA)

Quadratic discriminant analysis (QDA), first introduced by Smith (1947), is another distance-based classifier, which is very similar to the linear discriminant function classifier. In fact, quadratic discriminant analysis is an extended of the linear discriminant function. Both discriminant functions assume that the values of each attribute in each class are normally distributed, however, the discriminant score between each sample and each class is calculated using the sample variance–covariance matrix of each class separately rather than the overall pooled matrix and so is a method that takes into account the different variance of each class.

On the other hand, in linear discriminant analysis it is assumed that the covariance matrices of the groups are equal, whereas quadratic discriminant analysis makes no such assumption. When the covariance matrices are not equal, the boundary between the classes will be a hyper-conic and in theory, the use of quadratic discriminant analysis will result in better discrimination and classification rates. However, due to the increased number of additional parameters that need to be estimated, it is quite possible that the classification by quadratic discriminant analysis is worse than that of linear discriminant analysis (Malhotra et al. 1999). The quadratic discriminant is found by evaluating the equation:

$$\begin{aligned}&X'\left(S_1^{-1}-S_2^{-1}\right)X+2\left(\bar{X}_2'S_2^{-1}-\bar{X}_1'S_1^{-1}\right)X-\\&\qquad\left[\bar{X}_2'S_2^{-1}\bar{X}_2-\bar{X}_1'S_1^{-1}\bar{X}_1+Ln\left(|S_2|\big/|S_1|\right)\right]>c\end{aligned} \qquad (8)$$

The same conditions apply to the nature of c and classification in the case that the result is equal to c or zero. As with the linear discriminant, the quadratic discriminant function distinguishes between two classes. For multiple class data sets, this was handled the same as for linear discriminant analysis. The size of the differences in variances determines how much better the quadratic discriminant function will perform than the linear discriminant. For large variance differences, the quadratic discriminant excels when compared to the linear discriminant. Additionally, of the two, only the quadratic discriminant can be used when population means are equal. Although more broadly applicable than the linear discriminant, the quadratic discriminant is less resilient under non-optimal conditions. The quadratic discriminant can behave worse than the linear discriminant for small sample sizes. Additionally, data that is not normally distributed results in a poorer performance by the quadratic discriminant, when compared to the linear discriminant.

Marks and Dunn (1974) found the performance of the quadratic discriminant function to be more sensitive to the dimensions of the data than the linear discriminant, improving as the number of attributes increases to a certain optimal number, then rapidly declining. Linear and nonlinear discriminant functions are the most widely used classification methods. This broad acceptance is due to their ease of use and the wide availability of tools. Both, however, assume the form of the class boundary is known and fits a specific shape. This shape is assumed to be smooth and described by a known function. These assumptions may fail in many cases. In order to perform classification for a wider range of real-world data, a method must be able to describe boundaries of unknown, and possibly discontinuous, shapes.

K-Nearest Neighbour (KNN)

The *K*-nearest neighbour (KNN) model is a well-known supervised learning algorithm for pattern recognition that first introduced by Fix and Hodges in 1951, and is still one of the most popular nonparametric models for classification problems (Fix & Hodges 1951; 1952). *K*-nearest neighbour assumes that observations, which are close together, are likely to have the same classification. The probability that a point x belongs to a class can be estimated by the proportion of training points in a specified neighbourhood of x that belong to that class. The point may either be classified by majority vote or by a similarity degree sum of the specified number (*k*) of nearest points. In majority voting, the number of points in the neighbourhood belonging to each class is counted, and the class to which the highest proportion of points belongs is the most likely classification of x. The similarity degree sum calculates a similarity score for each class based on the *K*-nearest points and classifies x into the class with the highest similarity score. Due to its lower sensitivity to outliers, majority voting is more commonly used than the similarity degree sum (Chaovalitwongse, 2007). In this paper, majority voting is used for the data sets.

In order to determine which points belong in the neighbourhood, the distances from x to all points in the training set must be calculated. Any distance function that specifies which of two points is closer to the sample point could be employed (Fix & Hodges 1951. The most common distance metric used in *K*-nearest neighbour is the Euclidean distance (Viaene, 2002). The Euclidean distance between each test point f_t and training set point f_s, each with n attributes, is calculated using the equation:

$$d = \left[(f_{t1} - f_{s1})^2 + (f_{t2} - f_{s2})^2 + \ldots\ldots + (f_{tn} - f_{sn})^2\right]^{1/2} \tag{9}$$

In general the following steps are performed for the *K*-nearest neighbour model (Yildiz et al., 2008):

i) Chosen of *k* value.

ii) Distance calculation.

iii) Distance sort in ascending order.

iv) Finding *k* class values.

v) Finding dominant class.

One challenge to use the *K*-nearest neighbour is to determine the optimal size of *k*, which acts as a smoothing parameter. A small *k* will not be sufficient to accurately estimate the population proportions around the test point. A larger *k* will result in less variance in probability estimates but the risk of introducing more bias. *K* should be large enough to minimize the probability of a non-Bayes decision, but small enough that the points included give an accurate estimate of the true class. Enas and Choi (1986) found that the optimal value of k depends upon the sample size and covariance structures in each population, as well as the proportions for each population in the total sample. For cases in which the differences in the covariance matrices and the difference between sample proportions were either both small or both large, Enas and Choi (1986) found that the optimal k to be $N^{3/8}$, where N is the number of samples in the training set. When there was a large difference between covariance matrices and a small difference between sample proportions, or vice versa, they determined $N^{2/8}$ to be the optimal value of *k*. This model presents several advantages (Berrueta et al., 2007):

(i) Its mathematical simplicity, which does not prevent it from achieving classification results as good as (or even better than) other more complex pattern recognition techniques.

(ii) It is free from statistical assumptions, such as the normal distribution of the variables.

(iii) Its effectiveness does not depend on the space distribution of the classes.

In additional, when the boundaries between classes cannot be described as hyper-linear or hyper-conic, K-nearest neighbour performs better than the linear and quadratic discriminant functions. Enas and Choi (1986) found that the linear discriminant performs slightly better than *K*-nearest neighbour when population covariance matrices are equal, a condition that suggests a linear boundary. As the differences in the covariance matrices increases, *K*-nearest neighbour performs increasingly better than the linear discriminant function.

However, despite of the all advantages cited for the *K*-nearest neighbour models, they also have some disadvantages. *K*-nearest neighbour model cannot work well if large differences are present in the number of samples in each class. *K*-nearest neighbour provides poor information about the structure of the classes and of the relative importance of each variable in the classification. Furthermore, it does not allow a graphical representation of the results, and in the case of large number of samples, the computation can become excessively slow. In addition, *K*-nearest neighbour model much higher memory and processing requirements than other methods. All prototypes in the training set must be stored in memory and used to calculate the Euclidean distance from every test sample. The computational complexity grows exponentially as the number of prototypes increases (Muezzinoglu & Zurada, 2006).

Support Vector Machines (SVMs)

Support vector machines (SVMs) are a new pattern recognition tool theoretically founded on Vapnik's statistical learning theory (Vapnik, 1998). Support vector machines, originally designed for binary classification, employs supervised learning to find the optimal separating hyper-plane between the two groups of data. Having found such a plane, support vector machines can then predict the classification of an unlabeled example by asking on which side of the separating plane the example lies. Support vector machine acts as a linear classifier in a high dimensional feature space originated by a projection of the original input space, the resulting classifier is in general non-linear in the input space and it achieves good generalization performances by maximizing the margin between the two classes. In the following, we give a short outline of construction of support vector machine.

Consider a set of training examples as follows:

$$\{(x_i, y_i)\} \quad x_i \in R^n, y_i \in \{+1, -1\}; \quad i = 1,2,\ldots,m \tag{10}$$

where the x_i are real n-dimensional pattern vectors and the y_i are dichotomous labels. Support vector machine maps the pattern vectors $x \in R^n$ into a possibly higher dimensional feature space ($z = \phi(x)$) and construct an optimal hyper-plane $w \cdot z + b = 0$ in feature space to separate examples from the two classes. For support vector machine with L1 soft-margin formulation, this is done by solving the primal optimization problem as follows:

$$\begin{aligned} &Min \quad \frac{1}{2}\|w\|^2 + C\sum\nolimits_{i=1}^{m} \xi_i \\ &s.t. \quad y_i(w \cdot z_i + b) \geq 1 - \xi_i \quad \xi_i \geq 0, i = 1,2,\ldots,m \end{aligned} \tag{11}$$

where *C* is a regularization parameter used to decide a trade off between the training error and the margin, and ξ_i $(i = 1,2,\ldots,m)$ are slack variables. The above problem is computationally solved using the solution of its dual form:

$$\begin{aligned} &\underset{\alpha}{Max} \quad \sum_{i=1}^{m} \alpha_i - \frac{1}{2}\sum_{j=1}^{m} \alpha_i \alpha_j y_i y_j k(x_i, x_j) \\ &s.t. \quad \sum_{i=1}^{m} \alpha_i y_i = 0; \quad 0 \leq \alpha_i \leq C, i = 1,2,\ldots,m \end{aligned} \tag{12}$$

where $k(x_i, x_i) = \phi(x_i) \cdot \phi(x_j)$ is the kernel function that implicitly define a mapping ϕ. The resulting decision function is:

$$f(x) = sgn\left\{\sum_{i=1}^{m} \alpha_i y_i k(x_i, x) + b\right\} \tag{13}$$

All kernel functions have to fulfil Mercer theorem; however, the most commonly used kernel functions are polynomial kernel and radial basis function kernel, respectively (Song & Tang, 2005).

$$k(x_i, x_j) = (a(x_i, x_j) + b)^d \tag{14}$$

$$k\left(x_i, x_j\right) = exp\left(-g\left\|x_i, x_j\right\|^2\right) \tag{15}$$

Support vector machines differ from discriminant analysis in two significant ways. First, the feature space of a classification problem is not assumed to be linearly separable. Rather, a nonlinear mapping function (also called a kernel function) is used to represent the data in higher dimensions where the boundary between classes is assumed to be linear (Duda et al., 2001). Second, the boundary is represented by support vector machines instead of a single boundary. Support vectors run through the sample patterns which are the most difficult to classify, thus the sample patterns that are closest to the actual boundary. Over-fitting is prevented by specifying a maximum margin that separates the hyper plane from the classes. Samples, which violate this margin, are penalized. The size of the penalty is a parameter often referred to as *C* (Brown et al., 2000; Christianini & Taylor, 2000).

Formulation the Hybrid Proposed Model

Multi-layer perceptrons (MLPs) are flexible computing frameworks and universal approximators that can be applied to a wide range of classification problems with a high degree of accuracy (Khashei et al., 2012). Several distinguishing features of multi-layer perceptrons make them valuable and attractive for classification tasks. The most important of these, is that MLPs, as opposed to the traditional model-based techniques, are data-driven self-adaptive methods in that there are few a priori assumptions about the models for problems under study (Khashei et al., 2009). The parameter of MLP models (weights and biases) are crisp ($w_{i,j}\left(i=0,1,2,...,p \quad j=1,2,...,q\right)$, $w_j\left(j=0,1,2,...,q\right)$). In proposed model, instead of using crisp, fuzzy parameters in the form of triangular fuzzy numbers are used for related parameters of layers ($\tilde{w}_{i,j}\left(i=0,1,2,...,p \quad j=1,2,...,q\right)$, $\tilde{w}_j\left(j=0,1,2,...,q\right)$). The model is described using a fuzzy function with a fuzzy parameter (Khashei et al., 2012):

$$\tilde{y}_t = f\left(\tilde{w}_0 + \sum_{j=1}^{q} \tilde{w}_j \cdot g\left(\tilde{w}_{0,j} + \sum_{i=1}^{p} \tilde{w}_{i,j} \cdot y_{t-i}\right)\right), \tag{16}$$

Where, y_t are observations, $\tilde{w}_j\left(j=0,1,2,...,q\right)$, $\tilde{w}_{i,j}\left(i=0,1,2,...,p \quad j=1,2,...,q\right)$, are fuzzy numbers. Eq. (16) is modified as follows:

$$\tilde{y}_t = f\left(\tilde{w}_0 + \sum_{j=1}^{q} \tilde{w}_j \cdot \tilde{X}_{t,j}\right) = f\left(\sum_{j=0}^{q} \tilde{w}_j \cdot \tilde{X}_{t,j}\right), \tag{17}$$

where , $\tilde{X}_{t,j} = g\left(\tilde{w}_{0,j} + \sum_{i=1}^{p} \tilde{w}_{i,j} \cdot y_{t-i}\right)$. Fuzzy parameters in the form of triangular fuzzy numbers $\tilde{w}_{i,j} = \left(a_{i,j}, b_{i,j}, c_{i,j}\right)$ are used:

$$\mu_{\tilde{w}_{i,j}}\left(w_{i,j}\right) = \begin{cases} \frac{1}{b_{i,j} - a_{i,j}}\left(w_{i,j} - a_{i,j}\right) & if \quad a_{i,j} \le w_{i,j} \le b_{i,j}, \\ \frac{1}{b_{i,j} - c_{i,j}}\left(w_{i,j} - c_{i,j}\right) & if \quad b_{i,j} \le w_{i,j} \le c_{i,j}, \\ 0 & otherwise, \end{cases} \tag{18}$$

Where, $\mu_{\tilde{w}}\left(w_{i,j}\right)$ is the membership function of the fuzzy set that represents parameter $w_{i,j}$. By applying the extension principle, it becomes clear that the membership of $\tilde{X}_{t,j} = g\left(\tilde{w}_{0,j} + \sum_{i=1}^{p} \tilde{w}_{i,j} \cdot y_{t-i}\right)$ in Eq. (17) is given as (Khashei et al., 2008):

$$\mu_{\tilde{X}_{t,j}}\left(x_{t,j}\right) = \begin{cases} \frac{\left(X_{t,j} - g\left(\sum_{i=0}^{p} a_{i,j} \cdot y_{t,i}\right)\right)}{g\left(\sum_{i=0}^{p} b_{i,j} \cdot y_{t,i}\right) - g\left(\sum_{i=0}^{p} a_{i,j} \cdot y_{t,i}\right)} & if \quad g\left(\sum_{i=0}^{p} a_{i,j} \cdot y_{t,i}\right) \le X_{t,j} \le g\left(\sum_{i=0}^{p} b_{i,j} \cdot y_{t,i}\right), \\ \frac{\left(X_{t,j} - g\left(\sum_{i=0}^{p} c_{i,j} \cdot y_{t,i}\right)\right)}{g\left(\sum_{i=0}^{p} b_{i,j} \cdot y_{t,i}\right) - g\left(\sum_{i=0}^{p} c_{i,j} \cdot y_{t,i}\right)} & if \quad g\left(\sum_{i=0}^{p} b_{i,j} \cdot y_{t,i}\right) \le X_{t,j} \le g\left(\sum_{i=0}^{p} c_{i,j} \cdot y_{t,i}\right), \\ 0 & otherwise, \end{cases} \tag{19}$$

where, $y_{t,i} = y_{t-i} \quad (t = 1,2,...,k \quad i = 1,2,...,p)$, and $y_{t,i} = 1 \quad (t = 1,2,...,k \quad i = 0)$. Considering triangular fuzzy numbers, $\tilde{X}_{t,j}$ with membership function Eq. (19) and triangular fuzzy parameters $\tilde{w}_j$ will be as follows:

$$\mu_{\tilde{w}_j}(w_j) = \begin{cases} \dfrac{1}{e_j - d_j}(w_j - d_j) & if \quad d_j \le w_j \le e_j, \\ \dfrac{1}{e_j - f_j}(w_j - f_j) & if \quad e_j \le w_j \le f_j, \\ 0 & otherwise, \end{cases} \tag{20}$$

The membership function of $\tilde{y}_t = f(\tilde{w}_0 + \sum_{j=1}^{q} \tilde{w}_j \cdot \tilde{X}_{t,j}) = f(\sum_{j=0}^{q} \tilde{w}_j \cdot \tilde{X}_{t,j})$ is given as (21). Now considering a threshold level h for all membership function values of observations, the nonlinear programming is given as (22).

As a special case and to present the simplicity and efficiency of the model, the triangular fuzzy numbers are considered symmetric, output neuron transfer function is considered to be linear, and connected weights between input and hidden layer are considered to be of a crisp form. The membership function of y_t in the special case mentioned is transformed as follows:

$$\mu_{\tilde{Y}}(y_t) = \begin{cases} 1 - \dfrac{\left| y_t - \sum_{j=0}^{q} \alpha_j \cdot X_{t,j} \right|}{\sum_{j=0}^{q} c_j \left| X_{t,j} \right|} & for \quad X_{t,j} \ne 0, \\ 0 & otherwise, \end{cases} \tag{23}$$

Simultaneously, y_t represents the tth observation and h-level is the threshold value representing the degree to which the model should be satisfied by all the data points $y_1, y_2, ..., y_k$. A choice of the h value influences the widths of the fuzzy parameters:

$$\mu_{\tilde{Y}}(y_t) \ge h \qquad for\, t = 1,2,....,k, \tag{24}$$

$$\mu_{\tilde{Y}}(y_t) \cong \begin{cases} \dfrac{-B_1}{2A_1} + \left[\left(\dfrac{B_1}{2A_1} \right)^2 - \dfrac{C_1 - f^{-1}(y_t)}{A_1} \right]^{1/2} & if \quad C_1 \le f^{-1}(y_t) \le C_3, \\ \dfrac{B_2}{2A_2} + \left[\left(\dfrac{B_2}{2A_2} \right)^2 - \dfrac{C_2 - f^{-1}(y_t)}{A_2} \right]^{1/2} & if \quad C_3 \le f^{-1}(y_t) \le C_2, \\ 0 & otherwise, \end{cases} \tag{21}$$

where,

$$A_1 = \sum_{j=0}^{q} (e_j - d_j) \cdot \left(g\left(\sum_{i=0}^{p} b_{i,j} \cdot y_{t,i} \right) - g\left(\sum_{i=0}^{p} a_{i,j} \cdot y_{t,i} \right) \right),$$

$$B_1 = \sum_{j=0}^{q} \left(d_j \cdot \left(g\left(\sum_{i=0}^{p} b_{i,j} \cdot y_{t,i} \right) - g\left(\sum_{i=0}^{p} a_{i,j} \cdot y_{t,i} \right) \right) + g\left(\sum_{i=0}^{p} a_{i,j} \cdot y_{t,i} \right) \cdot (e_j - d_j) \right),$$

$$A_2 = \sum_{j=0}^{q} (f_j - e_j) \cdot \left(g\left(\sum_{i=0}^{p} c_{i,j} \cdot y_{t,i} \right) - g\left(\sum_{i=0}^{p} b_{i,j} \cdot y_{t,i} \right) \right),$$

$$B_2 = \sum_{j=0}^{q} \left(f_j \cdot \left(g\left(\sum_{i=0}^{p} c_{i,j} \cdot y_{t,i} \right) - g\left(\sum_{i=0}^{p} b_{i,j} \cdot y_{t,i} \right) \right) + g\left(\sum_{i=0}^{p} c_{i,j} \cdot y_{t,i} \right) \cdot (f_j - e_j) \right),$$

$$C_1 = \sum_{j=0}^{q} \left(d_j \cdot g\left(\sum_{i=0}^{p} a_{i,j} \cdot y_{t,i} \right) \right) \qquad C_2 = \sum_{j=0}^{q} \left(f_j \cdot g\left(\sum_{i=0}^{p} c_{i,j} \cdot y_{t,i} \right) \right), \qquad C_3 = \sum_{j=0}^{q} \left(e_j \cdot g\left(\sum_{i=0}^{p} b_{i,j} \cdot y_{t,i} \right) \right),$$

$$\text{Min} \qquad \sum_{t=1}^{k}\sum\nolimits_{j=0}^{q}\left(f_j \cdot g\left(\sum\nolimits_{i=0}^{p} c_{i,j}\cdot y_{t,i}\right)\right)-\left(d_j \cdot g\left(\sum\nolimits_{i=0}^{p} a_{i,j}\cdot y_{t,i}\right)\right)$$

$$\text{Subject.to} \quad \begin{cases} \dfrac{-B_1}{2A_1}+\left[\left(\dfrac{B_1}{2A_1}\right)^2-\dfrac{C_1-f^{-1}(y_t)}{A_1}\right]^{1/2} \le h \quad if \quad C_1 \le f^{-1}(y_t) \le C_3, \quad for \quad t=1,2,\ldots,k, \\ \dfrac{B_2}{2A_2}+\left[\left(\dfrac{B_2}{2A_2}\right)^2-\dfrac{C_2-f^{-1}(y_t)}{A_2}\right]^{1/2} \le h \quad if \quad C_3 \le f^{-1}(y_t) \le C_2, \quad for \quad t=1,2,\ldots,k, \end{cases} \tag{22}$$

The index *t* refers to the number of non-fuzzy data used for constructing the model. On the other hand, the fuzziness *S* included in the model is defined by:

$$S=\sum_{j=0}^{q}\sum_{t=1}^{k} c_j |w_j| |X_{t,j}|, \tag{25}$$

Where, w_j is the connection weight between output neuron and *j*th neuron of the hidden layer; $x_{t,j}$ is the output value of *j*th neuron of the hidden layer in the time *t*. Next, the problem of finding the parameters in the proposed method is formulated as a linear programming problem as follows:

$$\text{Minimize} \qquad S=\sum_{j=0}^{q}\sum_{t=1}^{k} c_j |w_j| |X_{t,j}|$$

$$\text{subject.to} \begin{cases} \sum\limits_{j=0}^{q}\alpha_j X_{t,j}+(1-h)\left(\sum\limits_{j=0}^{q} c_j |X_{t,j}|\right) \ge y_t & t=1,2,..,k \\ \sum\limits_{j=0}^{q}\alpha_j X_{t,j}-(1-h)\left(\sum\limits_{j=0}^{q} c_j |X_{t,j}|\right) \le y_t & t=1,2,..,k \\ c_j \ge 0 \qquad for\ j=0,1,\ldots,q. \end{cases} \tag{26}$$

Then, the data around the upper and lower bound of the proposed model, when model has outliers with a wide spread, are deleted in accordance with Ishibuchi's recommendations. In order to make the model to include all possible conditions, c_j has a wide spread when the data set includes a significant difference or outlying case. Ishibuchi and Tanaka (1988) suggest that the data around the model's upper and lower boundaries be deleted so that the fuzzy regression model can be reformulated. Final point is that the output of the proposed model is fuzzy and continuous, while our classification problem differs in that its output is discrete and nonfuzzy. Therefore, in order to apply the proposed model to classification, certain modifications to the model needed to be made. For this purpose, each class is firstly assigned a numeric value, and then the membership probability of the output in each class is calculated as follows:

$$P_A = 1-P_B = \frac{\int_{-\infty}^{m} f(x)dx}{\int_{-\infty}^{+\infty} f(x)dx} = 1-\frac{\int_{m}^{+\infty} f(x)dx}{\int_{-\infty}^{+\infty} f(x)dx} \tag{27}$$

where P_A and P_B are the membership probability of the class A and class B, respectively, and m is the mean of the class values. Finally, the sample is put in the class with which its output has the largest probability. In proposed model, due to this fact that output is fuzzy, it may be better to apply the large class values. The larger class values expand small differences in the output, helping the model to become more sensitive to variations in the input. For example, instead of using the $\{-1,+1\}$ or $\{0,+1\}$, the $\{-10,+10\}$ or $\{-100,+100\}$ are better to be used as class values (Khashei et al., 2012).

Pima Indian Diabetes Data Set

The Pima Indian Diabetes data set is collected by the National Institute of Diabetes and Digestive and Kidney Diseases and consists of diabetes diagnoses (positive or negative) and attributes of female patients who are at least 21 years old and of Pima Indian heritage (Golub et al. 1999). The eight attributes represent 1) the number of times pregnant, 2) the results of an oral glucose tolerance test, 3) diastolic blood pressure (mm Hg), 4) triceps skin fold thickness (mm), 5) 2-h serum insulin (micro U/ml), 6) body mass index (weight in kg/(height in m)^2), 7) diabetes pedigree function, and 8) age (year). The two-dimensional distribution of these two classes against

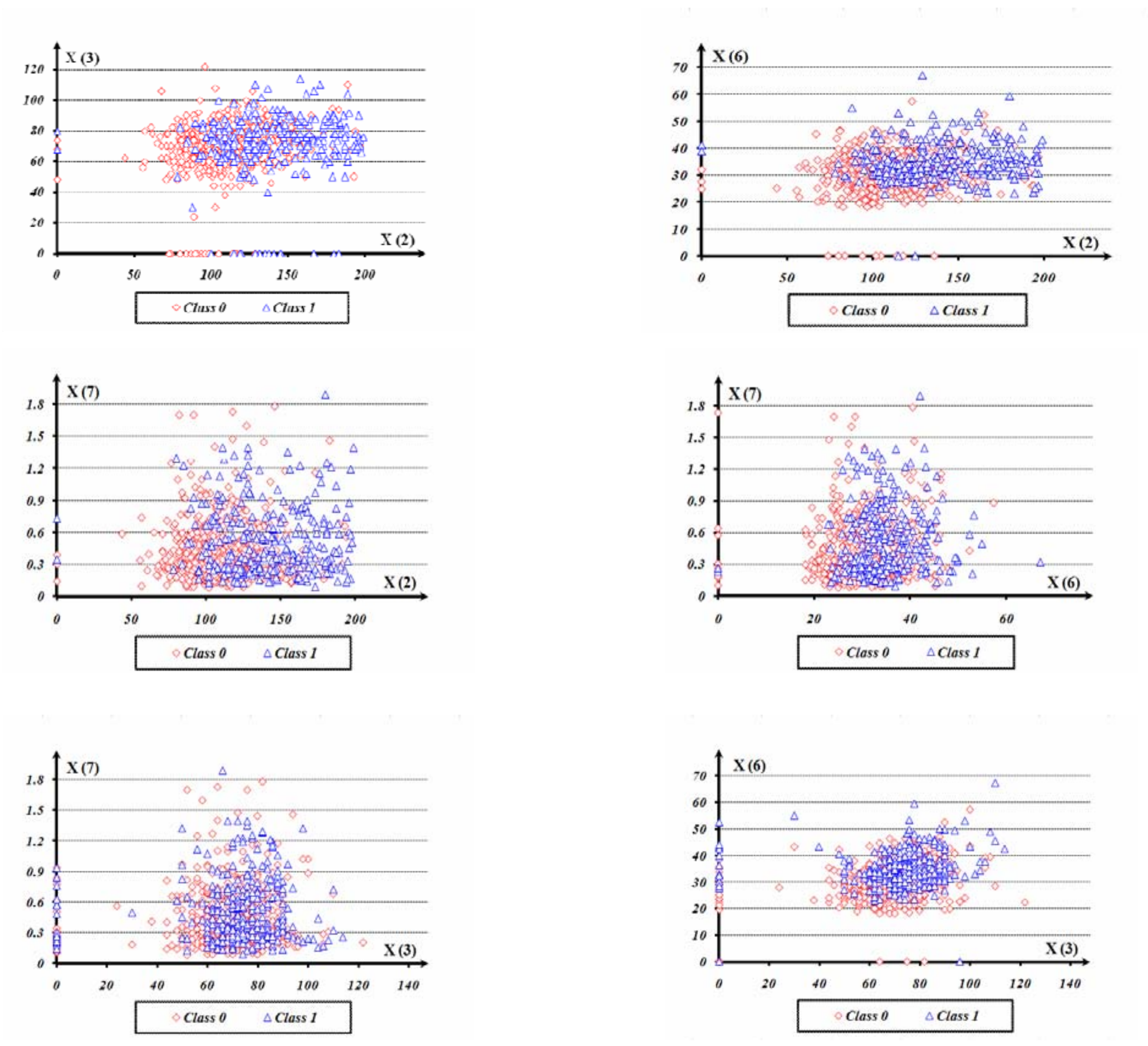

FIG. 2 THE TWO-DIMENSIONAL DISTRIBUTION OF PIMA INDIAN DIABETES CLASSES

TABLE 1 BRIEF STATISTICAL INFORMATION OF ATTRIBUTES

No.	Attribute Name	Mean	Standard Deviation
1-	Number of times pregnant	3.8	3.4
2-	Plasma glucose (2 Hours)	120.9	32
3-	Diastolic blood pressure	69.1	19.4
4-	Triceps skin fold thickness	20.5	16.0
5-	Two-hour serum insulin	79.8	115.2
6-	Body mass index	32.0	7.9
7-	Diabetes pedigree function	0.5	0.3
8-	Age	33.2	11.8

the (X2, X3), (X6, X7), (X2, X6), (X3, X7), (X2, X7), and (X3, X6), as example, is shown in Fig. 2. Some statistical information of attributes is given in Table 1. The data set consists of 768 samples, about two third of which have a negative diabetes diagnosis and one third with a positive diagnosis. The data set is randomly split into equal size of training and test sets of 384 samples each.

Application of the Hybrid Proposed Model to Diabetes Classification

In order to obtain the optimum network architecture of the proposed model based on the concepts of multi-layer perceptrons design (Khashei & Bijari, 2011) and using pruning algorithms in MATLAB 7 package software, different network architectures are evaluated to compare the MLPs performance.

The best fitted network which is selected, and therefore, the architecture which presented the best accuracy with the test data, is composed of eight inputs, five hidden and one output neurons (in abbreviated form, $N^{(8\text{-}5\text{-}1)}$). Then, the minimal fuzziness of the fuzzy parameters is determined using Eq. (11) with h=0.

As mentioned previously, the h-level value influences the widths of the fuzzy parameters. In this case, we consider h=0 in order to yield parameters with minimum of width. The misclassification rate of each model and improvement percentages of the proposed model in comparison with those of other classification models for the Pima Indian diabetes data in both training and test data sets are summarized in Table 2 and Table 3, respectively. The misclassification rate and improvement percentage of the model (B) against the model (A) are respectively calculated as follows:

$$Misclassification\ Rate\left(MR\right)=\frac{No.of\ \ incorrect\ diagnosis}{No.of\ \ sample\ set} \tag{28}$$

$$Improvement\ Percentage\ =\frac{MR\left(A\right)\text{-}\ MR\left(B\right)}{MR\left(A\right)}\times 100\% \tag{29}$$

Comparison with Other Models

According to the obtained results (Tables 2 & 3), our proposed model has the lowest error on the test portion of the data set in comparison to other those used models for the Pima Indian Diabetes data set, with a misclassification rate of 18.8%. Several different architectures of artificial neural network are designed and examined. The best performing architecture for a traditional multi-layer perceptron produces a 25.3% error rate, which proposed model improves by 25.69%. Linear discriminant analysis performs second best with an error rate of 21.9%, a classification rate 14.16% worse than the proposed model. Quadratic discriminant analysis misclassifies 28.1% of the test samples, which is also a 33.10% worse than the proposed model. As *K*-nearest neighbour scores can be sensitive to the relative magnitude of different attributes, all attributes are scaled by their z-scores before using *K*-nearest neighbour model (Antal et al., 2003). The best *K*-nearest neighbour, with a *K=13* has error rates of 24.7%that is a 23.89% higher than the proposed model error. The support vector machine model with *C=0* produces an error rate of 30.0%. The proposed model improves upon these by 37.33% for the support vector machine.

TABLE 2 PIMA INDIAN DIABETES DATA SET CLASSIFICATION RESULTS

Model	Classification error	
	Training Data	Test Data
Linear Discriminant Analysis (LDA)	*%26.6*	*%21.9*
Quadratic Discriminant Analysis (QDA)	*%23.7*	*%28.1*
K-Nearest Neighbour (KNN) [*K=13*]	*%23.4*	*%24.7*
Support Vector Machines (SVM) [*C=0*]	*%9.9*	*%30.0*
Artificial Neural Networks (ANN) [$N^{(8\text{-}5\text{-}1)}$]	*%18.8*	*%25.3*
Hybrid proposed model	*%17.6*	*%18.8*

TABLE 3 IMPROVEMENT OF THE PROPOSED MODEL IN COMPARISON WITH THOSE OF OTHER CLASSIFICATION MODELS

Model	Improvement (%)	
	Training Data	Test Data
Linear Discriminant Analysis (LDA)	*33.83*	*14.16*
Quadratic Discriminant Analysis (QDA)	*25.74*	*33.10*
K-Nearest Neighbour (KNN)	*24.79*	*23.89*
Support Vector Machines (SVM)	*-77.78*	*37.33*
Artificial Neural Networks (ANN)	*6.38*	*25.69*

Conclusions

Diabetes is a metabolic diseases characterized by high blood glucose levels, which result from body does not produce enough insulin or the body is resistant to the effects of insulin, named silent killer. Classification techniques have received considerable attention in biological and medical applications that greatly help physicians to improve their prognosis, diagnosis or treatment planning procedures. Both theoretical and empirical findings have indicated that using hybrid models or combining several models has become a common practice to reduce upon their misclassification rate, especially when the models in combination are quite different.

In this paper, a hybrid model of multi-layer perceptrons is proposed as an alternative classification model using the unique soft computing advantages of the fuzzy logic. The proposed model generally consists of five phases as follows:

i) Training the neural network using the available information from observations

ii) Determining the minimal fuzziness using the obtained weights and same criterion

iii) Deleting the outliers accordance with Ishibuchi's recommendations

iv) Calculating the membership probability of the output in each class

v) Assigning the output to appropriate class by the largest probability

Five well-known statistical and intelligent classification models —linear discriminant analysis, quadratic discriminant analysis, *K*-nearest neighbour, support vector machines, and multi-layer perceptrons—are used in this paper in order to show the appropriateness and effectiveness of the proposed model for diabetes classification. The obtained results indicate that the proposed model to be superior to all alternative models. For binary classification of the Pima Indian Diabetes benchmark data set, proposed model performs better than the traditional multi-layer perceptrons. The improvement varies from 6.38% to 25.69% in comparison to the multi-layer perceptrons for the training and test data sets. In addition, the performance of the hybrid proposed model is overall better than support vector machine and also other traditional classification models such as linear discriminant analysis and quadratic discriminant analysis.

Our proposed model does not assume the shape of the partition, unlike the linear and quadratic discriminant analysis. In contrast to the *K*-nearest neighbour model, the proposed model does not require storage of training data. Once the model has been trained, it performs much faster than *K*-nearest neighbour does, because it does not need to iterate through individual training samples. The proposed model does not require experimentation and final selection of a kernel function and a penalty parameter as is required by the support vector machines. Our proposed model solely relies on a training process in order to identify the final classifier model. Finally, the proposed model dose not need large amount of data in order to yield accurate results, as traditional multi-layer perceptrons.

ACKNOWLEDGEMENTS

The authors wish to express their gratitude to anonymous reviewers and Gh. A. Raissi Ardali, industrial engineering department, Isfahan University of Technology, who greatly helped us to improve this paper.

REFERENCES

Antal, P., Fannes, G., Timmerman, D., Moreau, Y., and Moor, B.D., "Bayesian applications of belief networks and multilayer perceptrons for ovarian tumor classification with rejection", Artificial Intelligence in Medicine Vol. 29, pp. 39– 60, 2003.

Benardos, P.G. and Vosniakos, G.C., "Optimizing feed-forward artificial neural network architecture", Engineering Applications of Artificial Intelligence, Vol. 20, pp. 365– 382, 2007.

Bennett, K. P. and Blue, J. A., "A support vector machine approach to decision trees", IEEE World Congress on Computational Intelligence, pp. 2396– 2401, 1998.

Berardi, V. and Zhang, G. P., "The effect of misclassification costs on neural network classifiers", Decision Sciences, Vol. 30, pp. 659– 68, 1999.

Berrueta, L., Alonso-Salces, R., Heberger, K., "Supervised pattern recognition in food analysis", Journal of Chromatography A, 1158, pp. 196– 214, 2007.

Billings, S. and Lee, K., "Nonlinear Fisher discriminant analysis using a minimum squared error cost function and the orthogonal least squares algorithm", Neural Networks, Vol. 15, pp. 262– 270, 2002.

Breault, J. L., Goodall, C. R., and Fos, P. J., "Data mining a diabetic data warehouse", Artificial Intelligence in Medicine, Vol. 26, pp. 37– 54, 2002.

Brown, M., Grundy, W., Lin, D., Cristianini, N., Sugnet, C., Furey, T., Ares, M., and Haussler, D., "Knowledge-based analysis of microarray gene expression data by using support vector machines" Proceedings of the National Academy of Sciences of the United States of America, Vol. 97, pp. 262– 267, 2000.

Calisir D. and Dogantekin, E., "An automatic diabetes diagnosis system based on LDA-Wavelet Support Vector Machine Classifier", Expert Systems with Applications, Vol. 38, pp. 8311– 8315, 2011.

Chaovalitwongse, W., "On the time series k-nearest neighbor classification of abnormal brain activity", IEEE Transactions on Systems, Man, and Cybernetics – Part A: Systems and Humans, Vol. 37, 2007.

Charya, S., Odedra, D., Samanta, S., and Vidyarthi, S., "Computational Intelligence in Early Diabetes Diagnosis:

A Review", The Review of Diabetes Studies, Vol. 7, pp. 252– 262, 2010.

Christianini, N.and Shawe-Taylor, J., "An introduction to support vector machines", Cambridge University Press, 2000.

Duda, R., Hart, P., and Stork, D., "Pattern classification", New York: John Wiley & Sons, Inc.2001.

Enas, G. and Choi, S., "Choice of the smoothing parameter and efficiency of k-nearest neighbor", Computers and Mathematics with Applications, Vol. 12, pp. 235– 244, 1986.

Fisher, R. A., "The use of multiple measurements in taxonomic problems", Annals of Eugenics, Vol. 7, pp. 465– 475, 1936.

Fix, E. and Hodges, J., "Discriminatory analysis – Nonparametric discrimination: Consistency properties", Project No. 21-49-004, Report No. 4, Contract No. AF 41(128)-31, USAF School of Aviation, Randolph Field, Texas, 1951.

Fix, E. and Hodges, J., "Discriminatory analysis– Nonparametric discrimination: Small sample performance. Project No. 21-49-004, Report No. 11, Contract No. AF 41(129)-31, USAF School of Aviation, Randolph Field, Texas, 1952.

Friedman, N., Geiger, D., and Goldszmit, M., "Bayesian networks classifiers", Machine Learning, Vol. 29, pp. 131– 163, 1997.

Ganji M. F. and Abadeh, M. S., "A fuzzy classification system based on Ant Colony Optimization for diabetes disease diagnosis", Expert Systems with Applications, Vol. 38, pp. 14650– 14659, 2011.

Ghiassi, M. and Burnley, C., "Measuring effectiveness of a dynamic artificial neural network algorithm for classification problems", Expert Systems with Applications, Vol. 37, pp. 3118– 3128, 2010.

Golub, T.R., Slonim, D.K., Tamayo, P., Huard, C., Gaasenbeek, M., Mesirov, J.P., Coller, H., Loh, M.L., Downing, J.R., Caligiuri, M.A., Bloomfield, C.D., and Lander, E.S., "Molecular classification of cancer: class discovery and class prediction by gene expression monitoring", Science, Vol. 286, pp. 531– 537, 1999.

Hosseini, H., Luo, D., and Reynolds, K.J., "The comparison of different feed forward neural network architectures for ECG signal diagnosis", Medical Engineering & Physics, Vol. 28, pp. 372– 378, 2006.

Huang, Y., McCullagh, P., Black, N., and Harper, R., "Feature selection and classification model construction on type 2 diabetic patients' data", Artificial Intelligence in Medicine, Vol. 41, pp. 251– 262, 2007.

Ishibuchi, H. and Tanaka, H., "Interval regression analysis based on mixed 0-1 integer programming problem", J. Japan Soc. Ind. Eng., Vol. 40, pp. 312– 319, 1988.

Jen, C. H., Wang, C. C., Jiang, B. C., Chu, Y. H., and Chen, M. S., "Application of classification techniques on development an early-warning system for chronic illnesses" Expert Systems with Applications, Vol. 39, pp. 8852– 8858, 2012.

Kahramanli H. and Allahverdi, N., "Design of a hybrid system for the diabetes and heart diseases", Expert Systems with Applications, Vol. 35, pp. 82– 89, 2008.

Kayaer, K. and Yildirim, T., "Medical diagnosis on pima Indian diabetes using general regression neural networks", artificial neural networks and neural information processing (ICANN/ICONIP), Istanbul, Turkey, pp. 181– 184, 2003.

Khashei, M. and Bijari, M., "A novel hybridization of artificial neural networks and ARIMA models for time series forecasting", Applied Soft Computing, Vol. 11, pp. 2664– 2675, 2011.

Khashei, M. and Bijari, M., "An artificial neural network (p, d, q) model for time series forecasting", Expert Systems with Applications, Vol. 37, pp. 479– 489, 2010.

Khashei, M., Bijari, M., and Hejazi, S. R., "Combining seasonal ARIMA models with computational intelligence techniques for time series forecasting", Soft Computing, Vol. 16, pp. 1091– 1105, 2012.

Khashei, M., Bijari, M., and Raissi, GH A., "Improvement of Auto-Regressive Integrated Moving Average models using fuzzy logic and artificial neural networks (ANNs)", Neurocomputing, Vol. 72, pp. 956– 967, 2009.

Khashei, M., Hamadani, A. Z., and Bijari, M., "A fuzzy intelligent approach to the classification problem in gene expression data analysis", Knowledge-Based Systems, Vol.27, pp. 465– 474, 2012.

Khashei, M., Hejazi, S. R., Bijari, M., "A new hybrid artificial neural networks and fuzzy regression model for time series forecasting", Fuzzy Sets and Systems, Vol. 159, pp. 769– 786, 2008.

Khashei, M., Zeinal Hamadani, A., Bijari, M., "A novel hybrid classification model of artificial neural networks and multiple linear regression models", Expert Systems with Applications, Vol. 39, pp. 2606– 2620, 2012.

Malhotra, M., Sharma, S. and Nair, S., "Decision making using multiple models", European Journal of Operational Research, Vol. 114, pp. 1– 14, 1999.

Marks, S. and Dunn, O., "Discriminant functions when covariance matrices are unequal", Journal of the American Statistical Association, Vol. 69, pp. 555– 559, 1974.

Muezzinoglu, M. and Zurada, J., "RBF-based neurodynamic nearest neighbor classification in real pattern space", Pattern Recognition, Vol. 39, pp. 747– 760, 2006.

Patil, B. M., Joshi, R. C., and Toshniwal, D., "Hybrid prediction model for Type-2 diabetic patients", Expert Systems with Applications, Vol. 37, pp. 8102– 8108, 2010.

Polat, K., Gunes, S., and Arslan, A., "A cascade learning system for classification of diabetes disease: Generalized discriminate analysis and least square support vector machine", Expert Systems with Applications, Vol. 34, pp. 482– 487, 2008.

Rumelhart, D. and McClelland, J., "Parallel distributed processing", Cambridge, MA: MIT Press, 1986.

Shi, Y., Eberhart, R., and Chen, Y., "Implementation of evolutionary fuzzy systems", IEEE Transactions on Fuzzy Systems, Vol. 7, pp. 109– 119, 1999.

Silva, L. Marques, J., and Alexandre, L. A., "Data classification with multilayer perceptrons using a generalized error function", Neural Networks, Vol. 21, pp. 1302– 1310, 2008.

Smith, C. A. "Some examples of discrimination", Annals of Eugenics, Vol. 13, pp. 272– 282. 1947.

Song, J. and Tang, H., "Support vector machines for classification of homo-oligomeric proteins by incorporating subsequence distributions", Journal of Molecular Structure: THEOCHEM 722, pp. 97– 101, 2005.

Su, C. T., Yang, C. H., Hsu, K. H., and Chiu, W. K., "Data mining for the diagnosis of type II diabetes from three-dimensional body surface anthropometrical scanning data", Computers & Mathematics with Applications, Vol. 51, pp. 1075– 1092, 2006.

Temurtas, H., Yumusak, N., and Temurtas, F., "A comparative study on diabetes disease diagnosis using neural networks", Expert Systems with Applications, Vol. 36, pp. 8610– 8615, 2009.

Vapnik, V., "Statistical learning theory", Wiley, New York, 1998.

Viaene, S., Derrig, R., Baesens, B., and Dadene, G., "A comparison of state-of-the art classification techniques for expert automobile insurance claim fraud detection", The Journal of Risk and Insurance, Vol. 69, pp. 373– 421, 2002.

Yildiz, T., Yildirim, S., Altilar, D., "Spam filtering with parallelized KNN algorithm", Akademik Bilisim, 2008.

Zadeh, L.A., "The concept of a linguistic variable and its application to approximate reasoning I", Information Sciences, Vol. 8, pp. 199– 249, 1975.

Zadeh, L.A., "The concept of a linguistic variable and its application to approximate reasoning II", Information Sciences, Vol. 8, pp. 301– 357, 1975.

Zhang, G. P., "An investigation of neural networks for linear time-series forecasting", Computers and Operations Research, Vol. 28, pp. 1112– 1183, 2001.

Zhang, G., Patuwo, B. E., and Hu, M. Y., "Forecasting with artificial neural networks: The state of the art", International Journal of Forecasting, Vol. 14, pp. 35– 62, 1998.

Zhao, H., "A multi-objective genetic programming approach to developing Pareto optimal decision trees", Decision Support Systems, Vol. 43, pp. 809– 826, 2007.